



## W&M ScholarWorks

---

School of Education Articles

School of Education

---

4-1990

## Reach Out and Touch Logo

Judi Harris

*College of William and Mary*

Follow this and additional works at: <https://scholarworks.wm.edu/educationpubs>

 Part of the [Education Commons](#)

---

### Recommended Citation

Harris, J. (1990). Reach out and touch Logo. Logo Exchange, 8(8), 15-17.

This Article is brought to you for free and open access by the School of Education at W&M ScholarWorks. It has been accepted for inclusion in School of Education Articles by an authorized administrator of W&M ScholarWorks. For more information, please contact [scholarworks@wm.edu](mailto:scholarworks@wm.edu).

## Logo LinX

### Reach Out and Touch Logo

by Judi Harris

Does paddling have a place in the Logo classroom? Certainly...when it is the computer's paddles that are being used. Paddle input is supported by most versions of Logo, and can be anything but painful.

Joysticks, game paddles, and touch-sensitive graphics tablets (such as the KoalaPad) are often used as alternative input devices for game and graphics programs. All are connected through a microcomputer's game port or game adaptor, and can be directly accessed with two Logo primitives.

With these inexpensive peripherals, pre-readers, physically challenged students, and those of us that just like to "piddle-paddle" can use Logo in a host of unique and exciting ways. Command of just two Logo commands may eliminate the need to purchase pre-programmed software that accesses touch tablets, and cannot be easily tailored to meet individual student needs.

In this article, I would like to concentrate on Logo interfaces and applications with graphics pads such as the KoalaPad, Animation Station, and Touch Window.

#### Touching Primitives

When someone touches a graphics pad connected to a microcomputer, two types of information can be detected and acted upon. The two-dimensional position of their finger on the surface of the tablet can be registered with the PADDLE command. The BUTTON? or BUTTONP (in Terrapin Logo, PADDLEBUTTON) command can also be used to determine if graphics pad buttons are being pressed.

Paddle information is typically represented by numbers ranging between 0 and 255. PADDLE 0 outputs position information along the X (horizontal) axis; PADDLE 1 numbers refer to Y (vertical) axis position. A simple recursive procedure can be used to print paddle information on the screen as you move your finger or a stylus over the surface of a graphics pad:

```
TO PADDLE.POS
CT
PRINT SENTENCE [PADDLE 0:] PADDLE 0
PRINT SENTENCE [PADDLE 1:] PADDLE 1
PADDLE.POS
END
```

Button information is output as either "TRUE or "FALSE; the former if the touch pad button indicated is being depressed, the latter if it is not.

```
TO BUTTON.PRESS?
CT
PRINT SENTENCE [BUTTON 0:] BUTTON? 0
PRINT SENTENCE [BUTTON 1:] BUTTON? 1
BUTTON.PRESS?
END
```

Paddle information is most commonly used to sense position of contact with the graphics tablet. Button information is typically used in a conditional statement that allows a user to select an option (such as a screen change or sound effect) whenever s/he chooses.

#### A Teacher's Touch

If position on the surface of the graphics tablet can be detected with Logo commands, why not correlate the position of the turtle on the screen with the location of the finger or stylus on the touch-sensitive pad?

At first glance, this seems simple enough:

```
SETPOS SENTENCE ( PADDLE 0 ) ( PADDLE 1 )
```

or, in Terrapin Logo

```
SETXY ( PADDLE 0 ) ( PADDLE 1 )
```

OOPS! PADDLE 0 inputs range from 0 to 255, but X-axis screen coordinates span approximately -140 to 140 or -120 to 120, depending on the version of Logo that is being used. If SETPOS is used with non-adjusted PADDLE 0 numbers, the turtle could only assume X axis positions between 0 and 255. Negative coordinate placements would be omitted, and screen boundaries would be ignored. There is a similar discrepancy with PADDLE 1 and Y-axis numbers.

Simple tool procedures that recalculate the range of paddle information and offset the turtle's screen position relative to sizes of different graphics pads can be used to correct the discrepancies.

For the KoalaPad:

```
TO X.POINT
OUTPUT (( PADDLE 0 ) - 131 ) * 1.078
END

TO Y.POINT
OUTPUT ((PADDLE 1 ) - 130 ) * - 0.975
END
```

**For Animation Station:**

```
TO X.POINT
OUTPUT (( PADDLE 0 ) - 128 ) * 1.025
END
```

```
TO Y.POINT
OUTPUT ((PADDLE 1 ) - 128 ) * - 0.86
END
```

**For the Touch Window:**

```
TO X.POINT
OUTPUT (( PADDLE 0 ) - 125 ) * 1.14
END
```

```
TO Y.POINT
OUTPUT ((PADDLE 1 ) - 120 ) * - .73
END
```

Placing the turtle at newly-calculated positions is a simple matter. Since screen coordinates are usually represented by integers (numbers without decimals), X.POINT and Y.POINT output should be simplified with the INT (integer) command (as in POINT, below) before setting the turtle's position to X.POINT and Y.POINT values with PLACE.TURTLE.

```
TO POINT
OUTPUT LIST ( INT X.POINT )
( INT Y.POINT )
END
```

```
TO PLACE.TURTLE
SETPOS POINT
PLACE.TURTLE
END
```

The turtle's screen position will now reflect changing points of contact on the graphics tablet.

**Touchy Areas**

X.POINT and Y.POINT information can also be used to delineate sensitive areas on the tablet. This is especially appropriate for Touch Window applications, since this type of graphics pad is translucent, and can be mounted on the front of a monitor, allowing users to see what is displayed on the screen through the graphics tablet itself.

Suppose that we wanted to divide the screen/tablet area into four sections, so that a physically impaired child would have to touch each screen section to see a picture displayed inside it. Four procedures could be written as follows:

```
TO SECT1?
OUTPUT AND ( X.POINT < 0 )
( Y.POINT > 10 )
END
```

```
TO SECT2?
OUTPUT AND ( X.POINT > 0 )
( Y.POINT > 10 )
END
```

```
TO SECT3?
OUTPUT AND ( X.POINT < 0 )
( Y.POINT < 10 )
END
```

```
TO SECT4?
OUTPUT AND ( X.POINT > 0 )
( Y.POINT < 10 )
END
```

These procedures could then be used as conditional input.

```
TO TOUCH.PICTURES
IF SECT1? [LOADPIC "UPPER.LEFT]
IF SECT2? [LOADPIC "UPPER.RIGHT]
IF SECT3? [LOADPIC "LOWER.LEFT]
IF SECT4? [LOADPIC "LOWER.RIGHT]
TOUCH.PICTURES
END
```

**A Touch of Creativity**

An interesting programming challenge might be to write a procedure that would accept four touches as the corners of a sensitive area, then automatically define a SECT? procedure. Sue Anderson, teacher of preschool handicapped children in Albemarle County, Virginia, conceived and solved this problem so that she could use a Touch Window in conjunction with a Logo-controlled videodisc player. Now when the speech synthesizer (also driven by Logo) tells her students to "touch the gorilla's belly," they can look through the clear graphics tablet at the videodisc image and touch the area that she defined with her time-saving tool procedure.

Other results for touching sensitive areas can be programmed easily. For example,

- Different musical notes could play when the appropriate lines or spaces were touched on a screen display of the musical staff.
- Maze walls could buzz when the turtle makes contact with them.
- Printed words (such as "cat," "truck," "blue," and "green")

could turn into the turtle shapes and colors that they describe, and be piloted around the screen.

- A display of piano keys could be used to sound out and automatically record single-note melodies.

The list of potential applications for a simple Logo interface seems endless.

### Keep in Touch

The power of this Logo-to-graphics-pad connection made itself apparent when I designed a 12-disk set of materials for a local teacher of young physically and mentally challenged people. I imported many pictures from *Print Shop* graphics, and made a set of Logo Touch Window programs in four levels of difficulty that help users to refine gross motor movements. You are welcome to use these public domain programs, too. Send 12 blank disks (or 6 double-notched blank disks) in a self-addressed, sufficiently stamped disk mailer to the address at the end of the article. I will copy the materials onto the disks and return them to you.

If you are curious about how to use *Print Shop* graphics in Logo programs, or Logo pictures in *Print Shop* creations, please refer to my September 1988 "Logo LinX" article, "Secular Conversions," in *Logo Exchange*.

If you are interested in learning more about controlling peripheral devices such as speech synthesizers and videodisc players with Logo, or accessing other input devices such as temperature or light sensors with Logo, the following articles may be helpful.

### References

- Barclay, T. (1987). To graph a probe—Logo in the science lab. *The Computing Teacher*, 14(8), 30-32.
- Bull, G., Bull, G., Lough, T., Harris, J. and Wissick, C. (1988). Logo connections: An open architecture for Logo. *Logo Exchange*, 6(8), 13-16.
- Bull, G. & Cochran, P. (1987). Science and sensors. *Logo Exchange*, 5(5), 9-11.
- Bull, G. & Cochran, P. (1987). Science and sensors II. *Logo Exchange*, 5(6), 9-11.
- Cochran, P. & Bull, G. (1986). Touch tools. *Logo Exchange*, 5(4), 18-20.
- Harris, J. (1988). Secular conversions. *Logo Exchange*, 7(1), pages undetermined at press time.

### Product Information

- "Animation Station," Suncom, Inc, \$99.95  
 "KoalaPad," Koala Technologies, \$125.00  
 "Touch Window," Personal Touch Corp., \$199.00

Note: A previous version of this article appeared as the November 1988 "Logo Center" column in *The Computing Teacher*.

Judi Harris  
 621F Madison Avenue  
 Charlottesville, VA 22903  
 CIS: 75116,1207  
 BitNet: JudiH@Virginia

## About the Cover

The perspective drawings to this month's cover were done by students of Orlando Mihich, a science teacher at Junior High School #118 in New York City. She writes

My kids got interested in perspective work after seeing my earth science drawings on the board, and after reviewing a book on Giorgio de Chirico. I explained the basic concepts of perspective drawing mainly during lunch hours and after school. Using the *seth towards* [], they drew lines looking towards a horizon, *make "x pos* helped them *setpos :x* for shading and so on. They came out with some quite interesting art work. The...sphere was earlier used to represent atoms in a crystal lattice; here it is used to add a round, soft shape to otherwise straight towers and pyramids. The distant LogoWriter train is the mysterious, metaphysical part of De Chirico's world itself.

DeChirico was done by Michael Toribio, 9th grade; Morning Meditation was done by John Deveau, 8th grade; and Yutz kin was done by Alex Acevedo, grade 9.